



Physikalische Chemie, Universität des Saarlandes

**Vorlesung: Programmierkonzepte in den Naturwissenschaften  
im Sommersemester 2021**

PD Dr. Till Biskup

— Glossar zu Lektion 29: „Datenformate: beständig und plattformunabhängig“ —

*Hinweis: Die nachfolgend genannten Begriffe und Definitionen erheben keinen Anspruch auf formale Korrektheit, sondern dienen lediglich dem besseren Verständnis der in der Vorlesung behandelten Themen und sind im jeweiligen Kontext zu sehen. Mehrfache, voneinander abweichende Definitionen in unterschiedlichen Kontexten sind daher möglich. Englische Begriffe werden zwar nach Möglichkeit übersetzt, erscheinen aber ggf. unter ihrem englischen Namen in der Liste. Verweise untereinander sind durch ↑ gekennzeichnet.*

**Abhängigkeit** *dependency*, im Quellcode durch explizite Nennung hervorgerufene ↑Kopplung von Programmteilen (↑Funktionen, ↑Objekte, ...), die dazu führt, dass der aufgerufene Programmteil nicht mehr ohne Veränderung des aufrufenden Teils verändert werden kann.

**Abstraktion** Nach Edsger Dijkstra das einzige mentale Werkzeug, das es erlaubt, eine große Vielzahl von Fällen abzudecken. Zweck der Abstraktion ist es nicht, vage zu sein, sondern im Gegenteil ein neues Bedeutungsniveau zu schaffen, das präzise Beschreibungen erlaubt.

**Abstraktionsebene** Summe aller ↑Abstraktionen eines bestimmten Abstraktionsgrades.

**ANSI** *American National Standards Institute*, private, gemeinnützige, amerikanische Organisation zur Koordinierung der Entwicklung freiwilliger Normen in den U.S.A.

**ASCII** *American Standard Code for Information Interchange*, 7-Bit-Zeichenkodierung, die als Grundlage für spätere, auf mehr Bits basierende Kodierungen für Zeichensätze dient. Die Zeichenkodierung definiert 128 Zeichen, bestehend aus 33 nicht druckbaren sowie 95 druckbaren Zeichen. Die druckbaren Zeichen umfassen das lateinische Alphabet in Groß- und Kleinschreibung, die zehn arabischen Ziffern sowie einige Interpunktionszeichen (Satz-

zeichen, Wortzeichen) und andere Sonderzeichen. Der Zeichenvorrat entspricht weitgehend dem einer Tastatur oder Schreibmaschine für die englische Sprache. Vgl. ↑Unicode, ↑UTF-8

**Attribut** im Kontext der ↑objektorientierten Programmierung eine Variable, die innerhalb einer ↑Klasse definiert wird. ↑Methoden operieren auf den Attributen einer ↑Klasse bzw. dem daraus erzeugten ↑Objekt.

**Auszeichnungssprache** *markup language* (ML) maschinenlesbare Sprache für die Gliederung und Formatierung von Texten und anderen Daten. Der bekannteste Vertreter ist die *Hypertext Markup Language* (↑HTML), die Kernsprache des World Wide Webs.

**Backup** ↑Datensicherung

**Clean Code** „sauberer Code“, letztlich lesbarer Code, der insbesondere im Kontext der naturwissenschaftlichen Datenauswertung die essentiellen Kriterien von Wiederverwendbarkeit, Zuverlässigkeit und Überprüfbarkeit erfüllt.

**CSV** *comma-separated values*, ↑Datenformat für zeilenweise Speicherung zusammengehöriger Daten, ähnlich der Zeilen in einer Tabelle. Die einzelnen Felder in einer Zeile werden oft durch Komma (daher der Name), ggf.

aber auch durch Semikolon getrennt. Im Gegensatz zu ↑DSV wurde CSV nie standardisiert, weshalb es Tabellenkalkulationen großer Hersteller gibt, die zwar CSV exportieren, aber den eigenen Export nicht mehr importieren können. Insbesondere der Umgang mit Feldtrennern innerhalb eines Feldes ist nicht festgelegt.

**Datenbank** *database* (DB) auch Datenbanksystem genannt; System zur elektronischen Datenverwaltung. Die wesentliche Aufgabe einer Datenbank besteht darin, große Datenmengen effizient, widerspruchsfrei und dauerhaft zu speichern und benötigte Teilmengen in unterschiedlichen, bedarfsgerechten Darstellungsformen für Benutzer und Anwendungsprogramme bereitzustellen. Eine Datenbank besteht aus zwei Teilen: der Verwaltungssoftware (Datenbankmanagementsystem, DBMS), und der Menge der zu verwaltenden Daten, der Datenbank (DB) im engeren Sinn. Die Verwaltungssoftware organisiert intern die strukturierte Speicherung der Daten und kontrolliert alle lesenden und schreibenden Zugriffe auf die Datenbank. Zur Abfrage und Verwaltung der Daten bietet ein Datenbanksystem eine Datenbanksprache (eine eigenständige Programmiersprache, häufig ↑SQL) an.

**Datenformat** digitales Speicherformat für Daten jeglicher Form. Grundsätzlich werden binäre und Textformate unterschieden. Während erstere meist mit deutlich geringerem Speicherbedarf auskommen, sind sie im Gegensatz zu letzteren nicht ohne Hilfsmittel lesbar. Textformate hingegen sind, ein beliebiger Texteditor vorausgesetzt, prinzipiell menschenlesbar. Wichtige Vertreter binärer Datenformate in den Naturwissenschaften sind ↑HDF5 und ↑IEEE 754 (eigentlich ein Standard für die Darstellung von Gleitkommazahlen). Wichtige Vertreter von Textformaten sind ↑CSV, ↑DSV, ↑JSON, ↑Windows-INI, ↑XML und ↑YAML.

**Datensicherung** *backup*, Kopieren von Daten in der Absicht, diese im Fall eines Datenverlustes zurückkopieren zu können, entsprechend eine

elementare Maßnahme zur Datensicherheit. Setzt eine sinnvolle Strategie zur langfristigen Absicherung sowohl gegen Verlust als auch gegen (ungewollte und oft zufällige) Veränderung voraus. Genauso entscheidend wie eine gute Strategie und regelmäßige Datensicherung ist dabei, die erzeugten Sicherungskopien regelmäßig auf korrekte Funktion überprüfen.

**DB** Abkürzung für ↑Datenbank

**Dependency Inversion** Umkehr der ↑Abhängigkeiten gegenüber der intuitiven Implementierung. Abhängigkeiten sollten häufig entgegen dem ↑Kontrollfluss verlaufen.

**Dependency-Inversion-Prinzip** (DIP) Anwendung der ↑Dependency Inversion: Abstraktionen sollten nicht von Details abhängen. Umgekehrt sollten Details auf Abstraktionen aufbauen.

**DSV** *delimiter-separated values*, ↑Datenformat für zeilenweise Speicherung zusammengehöriger Daten, ähnlich der Zeilen in einer Tabelle. Im Gegensatz zu ↑CSV ist das Format eindeutig definiert. Das Trennzeichen (*delimiter*) ist beliebig wählbar. Sollte es innerhalb eines Feldes auftauchen, wird es durch Backslash („\“) geschützt.

**Entität** allgemeine Bezeichnung für ein sprachliches bzw. gedankliches Objekt oder für ein außersprachliches Bezugsobjekt.

**Entwurfsmuster** *design patterns*, erprobte und bewährte Lösungen für wiederkehrende Probleme in der Softwareentwicklung. Beschreibungen miteinander kommunizierender ↑Objekte und ↑Klassen, die maßgeschneidert sind, um ein generelles Entwurfsproblem in einem bestimmten Kontext zu lösen. Entwurfsmuster liefern eine (↑abstrakte) Beschreibung des dahinterstehenden Konzepts und haben meist einen etablierten Namen, der die Kommunikation erleichtert. Es gibt ganze Kataloge solcher Muster, und viele der ursprünglich beschriebenen Entwurfsmuster sind heute in vielen Programmiersprachen fest etabliert.

**Framework** Rahmenstruktur, Ordnungsrahmen; in der Softwaretechnik ein Programmiergerüst,

das den Rahmen zur Verfügung stellt, innerhalb dessen der Programmierer eine Anwendung erstellt. Die Struktur der individuellen Anwendung wird u.a. durch die im Framework verwendeten ↑Entwurfsmuster beeinflusst. Das ↑Dependency-Inversion-Prinzip ist eine wesentliche Voraussetzung für den Entwurf flexibler, wiederverwendbarer Frameworks.

**freie Software** Software, die Endnutzern die Freiheiten der Nutzung, des Überprüfens/Studierens, des Teilens und des Modifizierens der Software gewährt. Das erfordert zwingend die Offenlegung des Quellcodes der Software. Der Begriff „frei“ bezieht sich hier auf die Freiheiten des Nutzers, nicht auf die Kosten der Software. Die Begriffe „open source“ und „freie Software“ sind nicht gänzlich deckungsgleich.

**Funktion** im Kontext der strukturierten Programmierung eine Liste von Anweisungen, die eine bestimmte Aufgabe erfüllt und der Programmiersprache unter einem festen Namen bekannt ist. Vgl. ↑Methode.

**größeres Projekt** hier: Alles, was mehr als zwei Wochen Arbeit kostet und deutlich mehr als zweihundert Zeilen (reinen) Quellcode bzw. mehr als eine Handvoll Unterfunktionen umfasst. Wichtig ist der Fokus: Sobald ein Programm über längere Zeit und/oder von anderen verwendet werden soll (was eher die Regel statt die Ausnahme ist), ist es ein größeres Projekt.

**Hash** ↑Prüfsumme

**HDF5** *Hierarchical Data Format*, ↑Datenformat, das insbesondere in wissenschaftlichen Anwendungen für die Speicherung großer Datenmengen verwendet wird. Optimierte Strukturen und Algorithmen erlauben das effiziente Speichern und Auslesen von ein- und mehrdimensionalen Tabellen, ohne dass jeweils der komplette Inhalt der Datei in den Speicher geladen werden muss. Tabellen und andere Daten können in ein und derselben Datei in einer beliebigen Verzeichnisstruktur abgelegt werden. Das ermöglicht u.a. die gleichzeitige

Speicherung von Messwerten und zugehörigen ↑Metadaten. Das Format wurde vom *National Center for Supercomputing Applications* (NCSA) entwickelt und wird u.a. von der NASA für Missionen verwendet.

**HTML** *Hypertext Markup Language*, textbasierte ↑Auszeichnungssprache zur Strukturierung elektronischer Dokumente wie Texte mit Hyperlinks, Bildern und anderen Inhalten. HTML-Dokumente sind die Grundlage des World Wide Web und werden von Webbrowsern dargestellt. Neben den vom Browser angezeigten Inhalten können HTML-Dateien zusätzliche Angaben in Form von Metainformationen enthalten, z. B. über die im Text verwendeten Sprachen, den Autor oder den zusammengefassten Inhalt des Textes.

**IEEE** *Institute of Electrical and Electronics Engineers*, weltweiter Berufsverband von Ingenieuren hauptsächlich aus den Bereichen Elektrotechnik und Informationstechnik. Der Verband veranstaltet Fachtagungen, gibt diverse Fachzeitschriften heraus und bildet Gremien für die Standardisierung von Techniken, Hardware und Software.

**IEEE 754** Norm des ↑IEEE, die die Standarddarstellungen für binäre Gleitkommazahlen in Computern definiert und genaue Verfahren für die Durchführung mathematischer Operationen, insbesondere für Rundungen, festlegt. In der Fassung ↑ANSI/IEEE 754-1985 ist nur der Standard für 64 Bit eindeutig.

**Importroutine** Softwareeinheit (oft eine ↑Funktion oder ↑Klasse), die ein ↑Datenformat einlesen und in eine für das jeweilige Programm intern sinnvolle Repräsentation umwandeln kann. Die Repräsentation von Daten innerhalb einer Anwendung sollte immer unabhängig vom jeweiligen externen Datenformat sein (eine Anwendung des ↑Dependency-Inversion-Prinzips).

**Infrastruktur** Personelle, sachliche und finanzielle Ausstattung, um ein angestrebtes Ziel zu erreichen. Im Kontext der Softwareentwicklung die Gesamtheit der Hilfsmittel, die (manche) Abläufe formalisieren und für Struktur und

Überprüfbarkeit sorgen. Erleichtert die Arbeit des Programmierers, indem sie viele Aspekte festlegt, die so zur Routine werden (und keine Denkleistung absorbieren).

**JSON** *JavaScript Object Notation*, hierarchisches ↑Datenformat, das ursprünglich zur einfachen Persistierung (↑Persistenz) von JavaScript-Objekten entwickelt wurde. Heute erfreut es sich als Austauschformat für Objekte und Datenstrukturen großer Beliebtheit, wird aber gerade für die Ablage von Konfigurationen in menschenlesbaren (und schreibbaren) Dateien aufgrund dessen noch einfacherer und übersichtlicherer Syntax zunehmend von ↑YAML abgelöst.

**Kapselung** *encapsulation*, ein ↑Objekt enthält Daten (↑Attribute) und zugehöriges Verhalten (↑Methoden) und kann beides nach Belieben vor anderen Objekten verstecken.

**Klasse** *class*, im Kontext der ↑objektorientierten Programmierung die Blaupause für die Erzeugung eines ↑Objektes; Definition der Daten (↑Attribute) und des zugehörigen Verhaltens (↑Methoden).

**Kollisionsresistenz** zu einer aus einer gegebenen Zeichenkette A errechneten ↑Prüfsumme existiert keine von A verschiedene Zeichenkette B, die zur gleichen Prüfsumme führt. Die Kollisionsresistenz einer ↑kryptographische Hashfunktion ist in der Praxis nie vollständig, aber ggf. hinreichend unwahrscheinlich.

**Kopplung** *coupling*, in Software der Grad der Verbindung zweier Komponenten; enge Bindung mehrerer Einheiten einer Software aneinander, so dass sie nicht unabhängig wiederverwendbar (bzw. ggf. auch nicht testbar) sind. Programmierkonzepte zielen generell auf eine lose Kopplung (*loose coupling*) einzelner Komponenten ab, da so die Wiederverwendbarkeit erleichtert wird.

**Kryptographie** ursprünglich die Wissenschaft der Verschlüsselung von Informationen. Heute befasst sie sich auch allgemein mit dem Thema Informationssicherheit, also der Konzeption, Definition und Konstruktion von Infor-

mationssystemen, die widerstandsfähig gegen Manipulation und unbefugtes Lesen sind.

**kryptographische Hashfunktion** Funktion, die eine Zeichenfolge beliebiger Länge auf eine solche fester Länge abbildet, ↑kollisionsresistent sein sollte und immer eine Einwegfunktion (↑unumkehrbar) ist.

**kryptographischer Hash** ↑Prüfsumme, die kryptographischen (↑Kryptographie) Gesichtspunkten entspricht bzw. sich Methoden der Kryptographie bedient und über eine ↑kryptographische Hashfunktion berechnet wird.

**Mehrfachvererbung** *multiple inheritance*, eine ↑Klasse erbt (↑Vererbung) von mehr als einer ↑Superklasse. Wird von den wenigsten Programmiersprachen unterstützt, oftmals behilft man sich hier aber des Konzeptes einer ↑Schnittstelle (*interface*) (3.) und kann dann mehr als eine solche implementieren (bzw. davon erben). Konzeptionell lassen sich diese beiden Ansätze quasi identisch einsetzen.

**Metadaten** Informationen zu den numerischen Daten, notwendige Voraussetzung für eine sinnvolle Verarbeitung der Daten im Kontext eines ↑Systems zur Datenverarbeitung und für ↑reproduzierbare Wissenschaft.

**Methode** im Kontext der ↑objektorientierten Programmierung eine ↑Funktion, die innerhalb einer ↑Klasse definiert wird und auf den ↑Attributen einer ↑Klasse bzw. dem daraus erzeugten ↑Objekt operiert.

**Modularisierung** Aufteilung der Gesamtaufgabe in kleinere Abschnitte. Die Aufteilung wird so lange fortgesetzt, bis die Lösung für den aktuellen Abschnitt unmittelbar in Form von Quellcode offensichtlich ist. Setzt die Definition von ↑Schnittstellen voraus.

**monolithisch** aus einem Stück bestehend; zusammenhängend und fugenlos

**Nutzerschnittstelle** ↑Schnittstelle zwischen Nutzer und Programm, meist auf hoher ↑Abstraktionsebene. Nutzerschnittstellen verstecken

für gewöhnlich die Implementierung und internen Abläufe einer Anwendung vor dem Nutzer. Typische Beispiele sind heute weit verbreitete grafische Oberflächen (*graphical user interface* GUI). Es gibt aber auch textbasierte (interaktive) Nutzerschnittstellen (*commandline interface*, CLI).

**Objekt** *object*, im Kontext der ↑objektorientierten Programmierung der grundlegende Baustein eines Programms, bestehend aus den Daten (↑Attribute) und dem zugehörigen Verhalten (↑Methoden). Ein Objekt ist in diesem Kontext immer die Instanz einer ↑Klasse.

**objektorientierte Programmierung** (OOP) ein ↑Programmierparadigma, bei dem Daten (Variablen zugewiesene Werte, als ↑Attribute bezeichnet) und Funktionen (↑Methoden), die auf diesen Daten (Attributen) operieren, eine Einheit bilden. Die in den ↑Attributen gespeicherten Daten lassen sich i.d.R. nur vermittelt durch (öffentlich zugängliche) ↑Methoden der ↑Klasse bzw. des daraus erzeugten ↑Objektes ansprechen. Es gibt eine klare Trennung zwischen öffentlicher ↑Schnittstelle und internen Verarbeitungsroutinen. Wichtige Vertreter objektorientierter Programmiersprachen sind Smalltalk, C++ und Java, aber auch Python.

**Paradigma** nach Thomas S. Kuhn ein Satz allgemein anerkannter wissenschaftlicher Leistungen, der für eine gewisse Zeit einer Gemeinschaft von Fachleuten maßgebende Probleme und Lösungen liefert

**Persistenz** Fähigkeit, Daten (oder ↑Objekte) oder logische Verbindungen über lange Zeit (insbesondere über einen Programmabbruch hinaus) bereitzuhalten; benötigt ein nichtflüchtiges Speichermedium.

**Persistenzschicht** ↑Schicht in der ↑Softwarearchitektur eines Gesamtsystems, deren Aufgabe die ↑Persistenz ist.

**Plattform** Zusammenspiel aus Betriebssystem und Hardware-Architektur. Betriebssysteme können bis zu einem gewissen Grad unterschiedliche Hardware-Architekturen abstrahieren, so

dass der gleiche Binärcode auf unterschiedlicher Hardware lauffähig ist, ohne neu kompiliert werden zu müssen.

**plattformunabhängig** Unabhängigkeit von einer spezifischen ↑Plattform. Neben Software sollten insbesondere ↑Datenformate plattformunabhängig sein, da nur so ein dauerhafter und unabhängiger Zugriff gewährleistet werden kann. Plattformunabhängigkeit hat sowohl eine Hardware- und Betriebssystemarchitektur als auch eine zeitliche Komponente. Alle drei großen Desktop-Betriebssysteme (Windows, Linux, macOS) sind in den Naturwissenschaften relativ weit verbreitet. Darüber hinaus findet die Entwicklung von Betriebssystemen auf der Zeitskala einer Archivierung schnell statt.

**Polymorphie** *polymorphism*, „Vielgestaltigkeit“, ähnliche ↑Objekte können auf die gleiche Botschaft (den Aufruf einer gleichnamigen ↑Methode) in unterschiedlicher Weise reagieren.

**Programmierparadigma** ein ↑Paradigma der Art zu programmieren. Wichtige Beispiele sind strukturierte Programmierung, ↑objektorientierte Programmierung und funktionale Programmierung.

**proprietär** auf herstellereigenen, (meist) nicht veröffentlichten Standards basierend

**Prüfsumme** *checksum, hash*, in der Informationstechnik ein Wert, der aus den Ausgangsdaten berechnet wurde und in der Lage ist, mindestens einen Bitfehler in den Daten zu erkennen. Ein einfaches Beispiel für eine Prüfsumme ist die Quersumme. Prüfsummen werden in Fehlerkorrekturmechanismen verwendet und lassen sich dazu verwenden, zufällige Veränderungen an Daten zu erkennen. Einfache Prüfsummen bieten aber keinerlei Schutz gegenüber absichtlichen Veränderungen. Dazu bedarf es ↑kryptographischer Hashes.

**quelloffen** Software, deren Quellcode offengelegt ist, so dass jeder ihn einsehen kann. Quelloffenheit ist eine notwendige, aber keine hinreichende Voraussetzung für ↑freie Software.

**redundant** mehrfach vorhanden

**reproduzierbare Wissenschaft** *reproducible science*, seit der Etablierung rechnergestützter Datenauswertung eigentlich nie mehr erreichter, aber für die Wissenschaft konstituierender Aspekt, dass sich Ergebnisse und Auswertungen unabhängig reproduzieren lassen, weil alle dazu notwendigen Aspekte vollständig und ausreichend beschrieben wurden. Motivation für die Vorlesung, deren Ziel es ist, die Hörer mit Konzepten vertraut zu machen, die letztlich eine ernstzunehmende reproduzierbare Wissenschaft ermöglichen.

**Schichten** *layer*, abstrakteste (↑Abstraktion) Organisationsebenen eines Gesamtsystems im Kontext der ↑Softwarearchitektur.

**Schnittstelle** *interface*, Begriff mit mehreren leicht unterschiedlichen Bedeutungen; (1.) ↑Signatur einer ↑Funktion oder ↑Methode. (2.) Im weiteren Sinne die Gesamtheit der öffentlichen ↑Attribute und ↑Methoden einer ↑Klasse bzw. eines ↑Objekts. Der Nutzer kennt nur die Schnittstelle, die Implementierung ist irrelevant und kann sich problemlos jederzeit ändern, solange die Funktionalität erhalten bleibt. Das dient der Trennung von Verantwortlichkeiten und ermöglicht ↑Modularisierung und ist in der Folge ein wesentlicher Aspekt der ↑Softwarearchitektur. (3.) In einer weiteren Bedeutung wird der Begriff (auch im Deutschen dann häufig mit seinem englischen Pendant) für (abstrakte) Klassen verwendet, die lediglich eine Schnittstelle (im Sinne von 2.) definieren. Das ist hauptsächlich dann von Bedeutung, wenn die Programmiersprache keine ↑Mehrfachvererbung unterstützt, aber das Implementieren von „*Interfaces*“.

**Schnittstellenadapter** in der ↑Softwarearchitektur eine ↑Abstraktionsebene, die zwischen der aus Sicht einer Anwendung externen Welt (↑Nutzerschnittstellen, ↑Persistenzschicht, ↑Datenbank etc.) und den in der Anwendung implementierten Anwendungsfällen vermittelt. Sorgt für eine ↑Modularisierung und Umkehr der Abhängigkeiten durch Einsatz des ↑Dependency-Inversion-Prinzips.

**Semantik** 1. Bedeutung bzw. Inhalt eines Wortes, Satzes oder Textes; 2. Teilgebiet der Linguistik, dessen Untersuchungsobjekt die Bedeutung sprachlicher Zeichen und Zeichenfolgen ist.

**Serialisierung** in der Informatik eine Abbildung von strukturierten Daten auf eine sequenzielle Darstellungsform. Serialisierung wird hauptsächlich für die Persistierung von Objekten in Dateien und für die Übertragung von Objekten über das Netzwerk bei verteilten Softwaresystemen verwendet.

**Softwarearchitektur** Aufteilung eines größeren Projektes in einzelne kleinere Projekte bzw. Aufgaben (↑Modularisierung), Definition klarer ↑Schnittstellen und Anforderungen sowie der Interaktion der einzelnen Teile miteinander. Nach Robert C. Martin die Gestalt eines Systems, die ihm von seinen Entwicklern gegeben wird: Unterteilung des Systems in Komponenten, ihre Anordnung, und die Art ihrer Interaktion miteinander.

**sprachunabhängig** hier: unabhängig von einer bestimmten Programmiersprache. ↑Datenformate sollten ↑quelloffen und in der Folge zumindest ↑Importroutinen sprachunabhängig implementierbar sein.

**SQL** *structured query language*, Datenbanksprache zur Definition von Datenstrukturen in relationalen ↑Datenbanken sowie zum Bearbeiten (Einfügen, Verändern, Löschen) und Abfragen von darauf basierenden Datenbeständen. Ziel der Verwendung von SQL ist die ↑Abstraktion von der verwendeten ↑Datenbank, auch wenn unterschiedliche Datenbanken oft leicht unterschiedliche Dialekte von SQL verwenden, was die Interoperabilität empfindlich einschränkt.

**Subklasse** ↑Klasse, die von einer anderen Klasse (der ↑Superklasse) ↑Attribute und ↑Methoden erbt. Die ↑Vererbung geht dabei i.d.R. über die nach außen hin sichtbare ↑Schnittstelle der Superklasse hinaus. Die Subklasse erbt von der ↑Superklasse häufig nur den „kleinsten gemeinsamen Nenner“ und implementiert die spezifische Funktionalität.

**Superklasse** ↑Klasse, von der andere Klassen (↑Subklassen) ↑Attribute und ↑Methoden erben. Die ↑Vererbung geht dabei i.d.R. über die nach außen hin sichtbare ↑Schnittstelle der Superklasse hinaus. Superklassen implementieren bzw. definieren normalerweise nur das Notwendigste, sozusagen den „kleinsten gemeinsamen Nenner“. Alle spezifische Funktionalität wird in der ↑Subklasse implementiert.

**syntaktisch validierbar** ↑Datenformat, das aufgrund seiner inneren Struktur hinsichtlich seiner korrekten Syntax validierbar ist. Meist wird für die Validierung ein Schema herangezogen, das die formale Beschreibung der Syntax in einer maschinenlesbaren Form enthält. Beispiele für syntaktisch validierbare Datenformate sind ↑XML und ↑HTML. Die syntaktische Validierbarkeit macht auch die Stärke dieser Datenformate aus, auch wenn für direkt menschenles- und schreibbare Dateien Formate mit deutlich geringerem Anteil an Steuerzeichen, z.B. ↑JSON und ↑YAML, bevorzugt werden.

**System zur Datenverarbeitung** hier: Gesamtsystem für wissenschaftliche Datenverarbeitung von der Datenaufnahme bis zur fertigen Publikation, das alle Aspekte umfasst und das ↑reproduzierbare Wissenschaft möglich macht und gewährleistet. Definitiv ein ↑größeres Projekt, das nicht nur eine ↑monolithische Anwendung umfasst, sondern viele Aspekte darüber hinaus. Setzt entsprechende ↑Infrastruktur und in der Umsetzung der einzelnen Komponenten sauberen Code (↑Clean Code) und eine solide ↑Softwarearchitektur voraus.

**Treiber** *driver*, Programm zur Kommunikation zwischen Soft- und Hardware, manchmal auch zwischen unterschiedlichen Programmen. Ein Treiber ist eine ↑Abstraktionsebene zwischen aufrufendem Programm und anzusteuern-der Einheit. Selbst wenn eine anzusteuern-der Hardware standardisierte Hardwareschnittstellen verwendet, ist der Satz an Befehlen, mit dem sie angesteuert werden kann, gerätespezifisch und wird entsprechend in einer Bibliothek codiert und als ↑Schnittstelle ande-

ren Programmen zur Verfügung gestellt. Auf Softwareebene kommen Treiber z.B. zur Ansteuerung von ↑Datenbanken zum Einsatz.

**Typisierung** *typing*, Zuweisung eines Typs zu einem Objekt (im abstrakten Sinne) einer Programmiersprache, z.B. Ganzzahl (*integer*) oder Zeichenkette (*string*) im Fall einer Variable. ↑Abstraktion, die die Ausdrucksstärke von Programmiersprachen und Programmen deutlich erhöht, und die Überprüfung der Korrektheit erleichtert sowie Optimierungen ermöglicht. Typisierung kann explizit und implizit erfolgen. Darüber hinaus wird zwischen starker und schwacher Typisierung sowie zwischen statischer und dynamischer Typisierung unterschieden. Jede Art der Typisierung hat ihre Vor- und Nachteile, und unterschiedliche Programmiersprachen verwenden unterschiedliche Arten der Typisierung.

**UCS** ↑*Universal Coded Character Set*

**UI** *User Interface*, vgl. ↑Nutzerschnittstelle

**Unicode** internationaler Standard, der langfristig für jedes Sinn tragende Schriftzeichen oder Textelement aller bekannten Schriftkulturen und Zeichensysteme einen digitalen Code festlegt. Dadurch soll die Verwendung unterschiedlicher, inkompatibler Kodierungen in verschiedenen Ländern oder Kulturkreisen beseitigt werden. Unicode wird ständig um Zeichen weiterer Schriftsysteme ergänzt. ISO 10646 ist die von ISO verwendete, praktisch bedeutungsgleiche Bezeichnung des Unicode-Zeichensatzes, der dort als ↑*Universal Coded Character Set* (UCS) bezeichnet wird. Vgl. ↑UTF-8

**Universal Coded Character Set** (UCS) Zeichenkodierung, die in der internationalen Norm ISO/IEC 10646 definiert ist und sich vollständig mit den korrespondierenden ↑Unicode-Kodierungen UTF-16 und UTF-32 deckt.

**unumkehrbar** im Kontext ↑kryptographischer Hashes die Tatsache, dass sich aus der Prüfsumme (*hash*) nicht die ursprüngliche Zeichenkette wiederherstellen lässt.

**UTF-8** *8-Bit ↑UCS Transformation Format*, die am weitesten verbreitete Kodierung für ↑Unicode-Zeichen (↑Unicode und ↑UCS sind praktisch identisch). UTF-8 ist in seinen ersten 128 Zeichen (Indizes 0–127) deckungsgleich mit ↑ASCII. Insbesondere englischsprachige Texte lassen sich deshalb im Regelfall ohne Modifikation sogar mit nicht-UTF-8-fähigen Texteditoren ohne Beeinträchtigung bearbeiten. Das ist einer der Gründe für den Status als *De-facto*-Standard-Zeichenkodierung des Internets und damit verbundener Dokumenttypen.

**Vererbung** *inheritance*, Weitergabe aller Eigenschaften (↑Attribute, ↑Methoden) von einer ↑Superklasse an eine ↑Subklasse. Die Subklasse ist vom gleichen Typ (↑Typisierung) wie die Superklasse, was wiederum die Grundlage der ↑Polymorphie ist. Änderungen der Superklasse wirken sich allerdings auf die Subklasse aus, die ↑Kapselung wird entsprechend geschwächt.

**Windows-INI** blockweise strukturiertes ↑Datenformat, das ursprünglich für die Ablage von Konfigurationsoptionen für das Windows-Betriebssystem und darauf laufende Programme entwickelt wurde. Die blockweise Struktur erlaubt zwei Hierarchieebenen: Blöcke und Schlüssel-Wert-Paare. Für weitere Hierarchieebenen und die Ablage beliebig verschachtelter Datenstrukturen müssen hierarchische Datenformate wie ↑XML, ↑JSON oder ↑YAML herangezogen werden.

**XML** *eXtensible Markup Language*, „erweiterbare Auszeichnungssprache“, ↑Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten im Format einer Textdatei, die sowohl von Menschen als auch von Maschinen lesbar ist. Der große Vorteil von XML ist seine ↑syntaktische Validierbarkeit, der Nachteil das Verhältnis von beschreibender Syntax zu eigentlichem Inhalt, das sowohl die Dateigröße erheblich erhöhen kann als auch die Lesbarkeit durch Menschen einschränkt. Alternativen, die von Menschen einfacher les- und insbesondere schreibbar sind, sind ↑JSON und ↑YAML.

**YAML** *YAML Ain't Markup Language* (rekursives Akronym, ursprünglich *Yet Another Markup Language*), vereinfachte Auszeichnungssprache (*markup language*) zur Datenserialisierung (↑Serialisierung). Die grundsätzliche Annahme von YAML ist, dass sich jede beliebige Datenstruktur nur mit assoziativen Listen, Listen (Arrays) und Einzelwerten (Skalaren) darstellen lässt. Durch dieses einfache Konzept ist YAML wesentlich leichter von Menschen zu lesen und zu schreiben als beispielsweise ↑XML, außerdem vereinfacht es die Weiterverarbeitung der Daten, da die meisten Sprachen solche Konstrukte bereits integriert haben. Durch weitgehenden Verzicht auf Klammern ist YAML für Menschen les- und insbesondere schreibbarer als ↑JSON und eignet sich daher gut für die Ablage von Metadaten und Konfigurationen.